



PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANTS: WRÖBEL -1

SERIAL NO.: 10/775,581

*Group 2122*

FILED: FEBRUARY 10, 2004

FOR: A METHOD OF SERVICING RECEPTION OF SOFTWARE  
BROADCASTED IN A CAROUSEL OF OBJECTS

CLAIM OF PRIORITY

MAIL STOP NON-FEE AMENDMENTS  
Commissioner for Patents  
P.O. BOX 1450  
Alexandria, VA 22313-1450

Dear Sir:

Applicant herewith claims the benefit of priority of his earlier-filed application under the International Convention in accordance with 35 U.S.C. 119. Submitted herewith is a certified copy and English translation thereof of the Polish application having the Serial No. P-358659, bearing the filing date of February 10, 2003.

It is hereby requested that receipt of this priority document be acknowledged by the Patent Office.

Respectfully submitted,  
WRÖBEL -1

*Edward Freedman*  
Allison C. Collard; Reg.No.22,532  
Edward R. Freedman; Reg.No.26,048  
Attorneys for Applicants

COLLARD & ROE, P.C.  
1077 Northern Boulevard  
Roslyn, New York 11576  
(516) 365-9802

Enclosures: Certified Copy of Polish Priority Document No. P-358659  
English Translation of Polish Priority Document No. P-358659

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. BOX 1450 Alexandria, VA 22313-1450, on ~~April 29, 2004~~

*Maria Guastella*  
Maria Guastella

*Jan. 5, 2005*



**ZAŚWIADCZENIE**

Advanced Digital Broadcast Ltd.  
Taipei, Tajwan

Advanced Digital Broadcast Polska Sp. z o.o.  
Zielona Góra, Polska

złożyli w Urzędzie Patentowym Rzeczypospolitej Polskiej dnia 10 lutego 2003 r. podanie o udzielenie patentu na wynalazek pt. „Sposób obsługi odbioru oprogramowania nadawanego w karuzeli obiektów”.

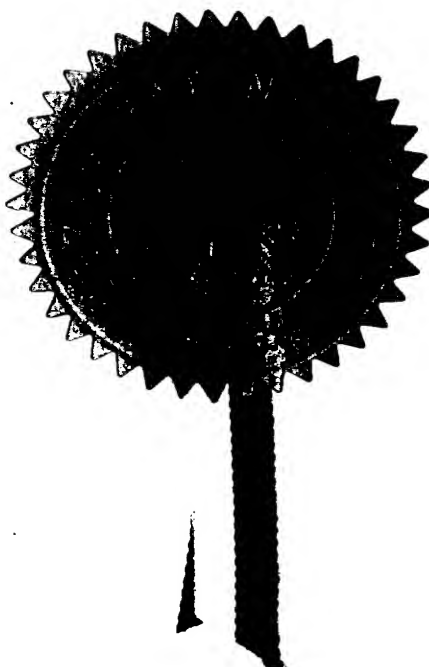
Dołączone do niniejszego zaświadczenia opis wynalazku, zastrzeżenia patentowe i rysunki są wierną kopią dokumentów złożonych przy podaniu w dniu 10 lutego 2003 r.

Podanie złożono za numerem P-358659.

Warszawa, dnia 11 lutego 2003 r.

z upoważnienia Prezesa

mgr Jowita Mazur  
Specjalista



**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

## **Sposób obsługi odbioru oprogramowania nadawanego w karuzeli obiektów**

Przedmiotem wynalazku jest sposób odbioru oprogramowania nadawanego w karuzeli obiektów, przeznaczonego korzystnie do zarządzania odbiorem oprogramowania w dekodernach telewizji cyfrowej, pracujących pod kontrolą systemu MHP – Multimedia Home Platform.

W znanym sposobie aktualizacji aplikacji w systemie MHP zakłada się wymianę danych tylko na żądanie programu, tzn. przez odwołanie aplikacji do brakującego modułu, po którym to żądaniu następuje wyszukiwanie odpowiedniego modułu w nadawanej w strumieniu karuzeli obiektów, przy czym należy podkreślić, że jest to proces bardzo czasochłonny.

Znany z amerykańskiego zgłoszenia patentowego US 2002/0059645 sposób zarządzania danymi karuzeli obiektów zakłada wprowadzenie modyfikacji po stronie nadajnika. Zapis obiektów nadawanych w karuzeli odbywa się na poziomie modułów aplikacji, przy czym w urządzeniu przechowuje się jedynie wersję modułu oraz czas, przez który dany moduł jest aktualny. Dodatkowo, jeśli wersje modułów się zmieniają, w odbiorniku przechowuje się wszystkie wersje oraz czas ich ważności. Wymóg modyfikacji strumienia danych po stronie nadajnika oraz gromadzenie w odbiorniku nadmiaru informacji czyni ten sposób bardzo skomplikowanym, a zatem czasochłonnym i niewygodnym w stosowaniu.

Istota sposobu według wynalazku polega na tym, że od momentu podłączenia odbiornika do nadającego jego oprogramowanie w postaci karuzeli obiektów strumienia danych, pobiera się należące do określonej karuzeli dane, w kolejności nadawania tych danych w strumieniu, a po odebraniu wszystkich danych tworzy się z nich moduły, które się dekompresuje, a następnie dołącza ich zawartość do systemu plików urządzenia.

Korzystnym jest gdy z pakietów strumienia danych pobiera się sekcję pakietu zawierającą moduły karuzeli, po czym sprawdza się czy sekcja zawiera potrzebne dane, które zapisuje się w pamięci urządzenia, sprawdzając przy tym kompletność pobranej karuzeli obiektów.

Ponadto korzystnym jest gdy pobierane ze strumienia danych sekcje danych, związane z określoną karuzelą obiektów, nadaje się w pakietach o określonym numerze PID.

Korzystnym jest także gdy odbiornik podłącza się do strumienia danych w odpowiedzi na żądanie działającej w odbiorniku aplikacji, określające numer PID pakietów, w których jest nadawana żądana przez aplikację karuzela obiektów.

Jednocześnie korzystnym jest gdy odbiornik podłącza się do strumienia danych w przypadku, gdy menedżer aplikacji chce uruchomić aplikację, która jest sygnalizowana w danym serwisie wraz z numerem PID pakietów, w których nadawana jest karuzela obiektów sygnalizowanej aplikacji.

Przedstawiony sposób umożliwia przechwytywanie informacji nadawanych cyklicznie w karuzeli obiektów oraz tworzenie systemu plików, z którego później korzystają aplikacje działające na przykład pod kontrolą systemu MHP. Modułów danych nie pobiera się jedynie w wyniku reakcji na odwołanie aplikacji do brakującego modułu, lecz również od razu po stwierdzeniu zmiany danych nadawanych w karuzeli obiektów. Rozwiązanie nie wymaga żadnych modyfikacji strumienia danych po stronie nadajnika oraz działa bez dodatkowych informacji dotyczących aktualności nadawanych modułów. Ponadto w odbiorniku przechowuje się tylko aktualną wersję modułu tzn. jego identyfikator, oraz sam moduł.

Przedmiot wynalazku został szczegółowo opisany w przykładzie, przy czym dla jego przybliżenia załączono rysunek, którego poszczególne figury przedstawiają:

- Fig.1 – sposób odbioru programu znany ze stanu techniki,
- Fig.2 – sposób odbioru programu według wynalazku,
- Fig.3 – sposób organizacji systemu plików za pomocą modułów,
- Fig.4 – porównanie czasu pobierania danych wykonywanego według znanego sposobu oraz sposobu według wynalazku,
- Fig.5 – wykorzystanie sposobu w przypadku gdy niektóre moduły nie zostaną wcześniej pobrane.

**P r z y k ł a d.** Nadawca wysyła oprogramowanie razem z sygnałem audio/video w jednym strumieniu danych. Odbiornik, którym może być dekodery telewizji cyfrowej IRD (Integrated Receiver Decoder), odbiera oprogramowanie, które następnie jest zapisywane w pamięci odbiornika, a potem uruchamiane lokalnie w urządzeniu. Przykładowo programy takie to elektroniczny przewodnik po programach EPG (Electronic Program Guide), gry czy serwisy informacyjne. Według aktualnego stanu techniki aplikacje nadawane są w postaci modułów w karuzeli obiektów, gdzie dane oraz kod programu nadawane są cyklicznie. Rozwiązanie takie podobne jest do systemu teletekstu, w którym strony nadawane są w cyklach.

Przykładowy system, w którym odbywa się transmisja w strumieniu danych telewizyjnych jest znany z międzynarodowego standardu ISO/IEC 13818-6 – MPEG-2 Digital Storage Media Command and Control – DSM-CC. Strumień MPEG zawiera multipleksowane kanały telewizyjne. Każdy taki kanał telewizyjny może zawierać interaktywne aplikacje przeznaczone dla odbiornika.

Obiekty przesyłane za pomocą protokołu DSM-CC są częścią karuzeli obiektów. Nadaje się je w postaci modułów, które tworzą system obiektów, plików. Organizacja systemu plików przechowywanych po stronie odbiornika jest podobna do systemów

znanych ze standardowych komputerów PC. Gdy odbiornik, na przykład dekodery telewizji cyfrowej, chce pobrać obiekt z karuzeli obiektów, musi wiedzieć, w którym module dany obiekt się znajduje. Po pobraniu odpowiedniego modułu przez odbiornik, moduł ten poddaje się analizie, tak aby znaleźć właściwy obiekt. Organizacja modułów została przedstawiona na **Fig. 3B**, gdzie trzy moduły zawierają system plików opisany na **Fig. 3A**. Każdy moduł (301) może zawierać pliki (303) i katalogi (302). Jeśli moduł zawiera plik, to musi także zawierać jego zawartość (305), zaś w przypadku katalogu moduł zawiera informację (304) o plikach/katalogach, które się w nim znajdują. Informacja ta zawiera nazwę pliku/katalogu oraz między innymi informację o module, w którym jest nadawany ten plik/katalog.

W związku z hierarchią obiektów nadawanych w karuzeli, dany obiekt może znajdować się w jednym z podkatalogów. W takim przypadku, aby znaleźć obiekt, odbiornik musi pobrać także wszystkie moduły nadrzędne względem modułu przechowującego obiekt, które muszą zostać przeanalizowane zanim odbiornik odczyta właściwy obiekt. Przykładowo, jeśli odbiornik potrzebuje pobrać obiekt „Plik” znajdujący się na ścieżce „/K1/K2/Plik”, musi najpierw sprawdzić, w jakim module jest nadawany obiekt "K1", następnie pobrać obiekt "K1", odczytać w jakim module jest nadawany "K2", ewentualnie pobrać ten moduł, i z obiektu "K2" odczytać w jakim module nadawany jest obiekt "Plik". W najprostszym przypadku wszystkie trzy obiekty mogą być w jednym module i wtedy wystarczy pobrać tylko ten jeden moduł. W najbardziej skomplikowanym przypadku, tak jak to przedstawiono na **Fig. 3B**, każdy z tych obiektów jest w innym module i aby sprawdzić, czy ścieżka jest poprawna i obiekt "Plik" istnieje, trzeba pobrać ze strumienia danych wszystkie trzy moduły.

Zazwyczaj wysyła się karuzelę obiektów w skompresowanej formie. Kompresja jest dodawana na poziomie modułów. Wynika z tego dodatkowe utrudnienie dla odbiornika, czyli wymóg dekompresji wszystkich modułów wymaganych do znalezienia obiektów, których w danej chwili wymaga odbiornik. Pojedynczy moduł zawiera zwykle wiele obiektów, które są skompresowane jako całość. System taki, biorąc pod uwagę możliwości przetwarzania danych w odbiornikach, na przykład dekoderych telewizji cyfrowej, jest czasochłonny i wymaga dużo zasobów systemu.

Dodatkowym utrudnieniem jest fakt, że pliki i katalogi systemu karuzeli obiektów mogą się zmieniać w czasie. W systemie transmisji danych DSM-CC ustala się wersje jedynie na poziomie modułów, a nie na poziomie hierarchii katalogów czy samych obiektów. W przypadku, gdy zmieni się tylko jeden obiekt z modułu, cały moduł otrzyma nową wersję i musi zostać ponownie pobrany w całości z karuzeli obiektów.

Kolejnym utrudnieniem jest fakt, że obiekty mogą być przenoszone pomiędzy poszczególnymi modułami. Można je także dodawać lub usuwać. Sytuacja taka prowadzi do dość częstych wymian obiektów będących częścią oprogramowania odbiornika, na przykład dekodera telewizji cyfrowej. W przedstawionym sposobie obiekt o zmienionej lokalizacji zostanie pobrany szybciej.

Według typowego sposobu odbioru danych, aby dołączyć karuzelę obiektów do systemu plików odbiornika, typowo dekodera telewizji cyfrowej, należy najpierw pobrać moduł z głównym elementem karuzeli. Następnie pobiera się kolejne moduły, zależnie od tego, których obiektów potrzebuje aplikacja. Typowy algorytm pobierania danych, wygląda tak jak zostało to przedstawione na **Fig. 1**. Po podłączeniu do

strumienia danych (101), pobiera się informacje o module głównym (ang. root), w punkcie (102) procedury. Następnie pobiera się informacje o liście modułów nadawanych w karuzeli (103). Po analizie tych informacji należy poczekać aż zostanie nadany moduł główny (104). Kolejnym punktem jest pobranie głównego modułu karuzeli obiektów (105). Następnym krokiem procedury jest odczyt innych potrzebnych dla aplikacji modułów (106). Ostatnim zadaniem procedury jest dołączenie pobranych obiektów do systemu plików (107), który jest zapisany w pamięci odbiornika (zwykle jest to pamięć typu FLASH), na przykład dekodera telewizji cyfrowej.

Chcąc określić czas potrzebny do prawidłowego pobrania aktualizacji dla aplikacji, która zawiera pięć modułów, należy wykonać poniższe wyliczenia:

- zakłada się, że pobranie informacji o module głównym (root) karuzeli trwa czas  $t_{\text{root}}$ ,
- zakłada się, że średni czas oczekiwania na informacje o module (root) karuzeli trwa czas  $t_{\text{root\_czekaj}}$ ,
- zakłada się, że pobranie informacji o liście modułów trwa czas  $t_{\text{lista}}$ ,
- zakłada się, że średni czas oczekiwania na informacje o liście modułów trwa  $t_{\text{lista\_czekaj}}$ ,
- zakłada się, że pobranie jednego modułu trwa czas  $t_{\text{modul}}$ ,
- zakłada się, że średni czas oczekiwania aż ten moduł zacznie być nadawany, wynosi  $t_{\text{modul\_czekaj}}$ .

W przypadku gdy aplikacja potrzebuje do działania  $N$  modułów, całkowity czas pobierania modułów dla tej aplikacji wynosi:

$$t_{\text{root}} + t_{\text{root\_czekaj}} + t_{\text{lista}} + t_{\text{lista\_czekaj}} + N * (t_{\text{modul\_czekaj}} + t_{\text{modul}})$$

W proponowanym sposobie zakłada się, że aplikacja w większości przypadków potrzebuje wszystkich modułów, które są nadawane w karuzeli obiektów. Ponadto karuzela jest nadawana cyklicznie, bez większych przerw pomiędzy jej końcem i początkiem.

Skoro aplikacja zwykle potrzebuje wszystkich modułów z karuzeli obiektów, to nie jest korzystnym czekać do momentu aż program odwoła się do nieistniejącego w systemie modułu, tylko należy pobrać wszystkie moduły od razu. W takim przypadku eliminuje się czas oczekiwania na poszczególne informacje, i zaczyna się od razu pobierać wszystkie moduły od momentu podłączenia odbiornika do strumienia danych. Zamiast pobierać nadawane moduły w odpowiedniej kolejności, pobiera się wszystkie moduły w kolejności, w której są one nadawane w karuzeli obiektów, licząc na to, że są to właściwe dane i informacja o nich zostanie nadana w najbliższym czasie. Może się więc zdarzyć, że w pierwszej kolejności pobrane zostaną moduły a następnie informacja o nich. Czas pobierania tej samej aplikacji, przy założeniu, że w karuzeli jest  $N$  modułów, wynosi:  $N * t_{\text{modul}}$ .

Stąd średni zysk w czasie wynosi:

$$t_{\text{root}} + t_{\text{root\_czekaj}} + t_{\text{lista}} + t_{\text{lista\_czekaj}} + N * t_{\text{modul\_czekaj}}$$

Informację o liście modułów i głównym module (root), nadaje się między modułami ze znacznie większą częstotliwością, dlatego do czasu pobierania aplikacji, używając sposobu według wynalazku, nie trzeba wliczać czasów  $t_{\text{root}} + t_{\text{root\_czekaj}} + t_{\text{lista\_czekaj}} + t_{\text{lista}}$ . Jednak w niektórych przypadkach sposób według wynalazku może mieć gorsze wyniki dotyczące czasu pobierania modułów. Na przykład w przypadku gdy w karuzeli jest dużo modułów a aplikacja potrzebuje tylko jeden.

Istnieje także ryzyko, że niektóre z modułów nie zostaną pobrane wcześniej i w takim przypadku pozostałe wymagane moduły są pobierane według typowego sposobu. Sytuacja taka została przedstawiona na Fig. 5. W punkcie (501) uruchamia się menedżera aplikacji, który zarządza programami danego urządzenia. Po uruchomieniu, monitoruje się dane przesyłane w strumieniu (502), na przykład formatu MPEG2. Kolejno sprawdza się czy w strumieniu jest sygnalizowana aplikacja (503). Jeśli nie ma takiego sygnału, dalej monitoruje się dane przesyłane w strumieniu. W przeciwnym wypadku pobiera się informacje określające parametry nadawanej w strumieniu aplikacji (504). Następnym krokiem procedury (505) jest pobranie karuzeli obiektów zgodnie ze sposobem według Fig. 2. Jeśli w czasie pobierania danych wystąpi błąd (506), moduły pobiera się w kroku (507) z karuzeli według znanych sposobów - Fig. 1. Do błędów należą sytuacje gdy odbiornik ma za mało pamięci na zapis całej karuzeli obiektów. Kolejnymi problemami są na przykład: spadek jakości sygnału lub sytuacja, gdy podczas pobierania danego modułu zmieni się jego wersja.

Pobieranie danych jest wykonywane bez przerwy aż do momentu, gdy odbiornik stwierdzi, że wszystkie dane z karuzeli nadawanej w pakietach strumienia o określonym numerze PID zostały już pobrane. Numer PID określa się przez aplikację, która żąda odczytu danych z karuzeli, nadawanej w pakietach strumienia podstawowego (ang. Elementary stream) o określonym numerze PID. Przez ciągłe pobieranie rozumie się to, że aktywuje się filtr sekcji DSM-CC pakietów MPEG2 tak, aby przepuszczał wszystkie możliwe dane związane z karuzelą nadawaną w pakietach o określonym numerze PID. Dopiero później, kompletuje się z pobranych danych moduły i pozostałe informacje. Gdyby dane były pobierane moduł po module, należałoby często ustawiać filtr na kolejne moduły, a później wyłączać go i włączać na nowo, aby pobrać kolejny moduł. Przez ten czas gdy wyłącza się filtr, moduł, który miał być pobrany, może już zostać nadany, co wiązałoby się z oczekiwaniem do kolejnego nadania karuzeli obiektów. Dodatkowo, ponieważ nie jest znana kolejność nadawania modułów, nigdy nie ma pewności, że zostałyby one pobrane w odpowiedniej, optymalnej kolejności. Sposób według wynalazku eliminuje te problemy oraz straty czasu dzięki temu, że pobiera się wszystkie sekcje danych, tak jak są one nadawane w karuzeli, a następnie sprawdza się ich rzeczywistą zawartość. Sposób pobierania danych ze strumienia, przykładowo w formacie MPEG-2, został przedstawiony na Fig. 2. Procedura rozpoczyna w momencie podłączenia do strumienia danych (201), które może być wynikiem żądania ze strony aplikacji. Następnie pobiera się sekcje DSM-CC z pakietów o określonym numerze PID (202). Kolejnym etapem (203) jest sprawdzenie czy pobrana sekcja danych zawiera moduł. Jeśli tak, następuje przejście do punktu (204) procedury. W przeciwnym wypadku następuje powrót do punktu (202), gdzie odbiera się kolejną sekcję ze strumienia. W

miejscu oznaczonym numerem (204) zapisuje się dane w pamięci urządzenia. Następnie sprawdza się czy pobrano już wszystkie sekcje danych (205), nadawanych w karuzeli obiektów z użyciem protokołu DSM-CC. Jeśli nie, procedura wraca do punktu (202). W przeciwnym wypadku, z pobranych danych utworzy się moduły (206). Kolejnym krokiem jest dekompresja modułów (207), tak aby w efekcie otrzymać właściwe obiekty nadawane w karuzeli. Ostatnim zadaniem procedury jest dołączenie pobranych obiektów do systemu plików (208), który jest zapisany w pamięci odbiornika - zwykle jest to pamięć typu FLASH, na przykład dekodera telewizji cyfrowej.

Jak pokazują testy praktyczne, sposób według wynalazku gwarantuje średnio dwukrotnie szybsze dostarczenie nowego modułu dla aplikacji działającej w odbiorniku, na przykład dekodерze telewizji cyfrowej. Przykładowe porównanie czasu potrzebnego na pobranie składającej się z trzech modułów karuzeli obiektów zostało przedstawione na Fig. 4. Część A przedstawia sposób znany ze stanu techniki, zaś część B - sposób według wynalazku. Jeden cykl karuzeli obiektów został oznaczony numerem (401). Podłączenie do strumienia danych następuje w miejscu oznaczonym numerem (402). Postępując zgodnie ze sposobem A, należy poczekać aż w strumieniu pojawi się moduł główny, który jest oznaczony symbolem „Moduł 1”. Czas ten oznaczony jest numerem (403). W punkcie (404), rozpoczyna się pobieranie modułu głównego, które trwa przez czas określony numerem (405). Czas (406) oznacza opóźnienie wynikające z analizy modułu głównego przez aplikację. W związku z tym opóźnieniem pobieranie modułu drugiego rozpoczyna się już podczas gdy część „Modułu 2” została nadana. Dlatego, aby pobrać resztę „Modułu 2”, należy poczekać aż zostanie on nadany ponownie. Czas pobierania „Modułu 2” określa zakres (407). Jako ostatni pobierany jest „Moduł 3”, w czasie oznaczonym numerem (408). Cała operacja pobrania trzech modułów, według znanego sposobu, trwa przez czas oznaczony numerem (409). Według wynalazku, moduły pobiera się ze strumienia kolejno, tak jak są nadawane. Od razu po podłączeniu do strumienia danych pobiera się moduły do momentu aż wszystkie moduły zostaną pobrane. Całą operacją przebiega w czasie oznaczonym jako (410). Dopiero później następuje ich analiza. Dzięki temu czas potrzebny na aktualizację danych zostaje zredukowany do niezbędnego minimum oraz aplikacje działające w odbiorniku mają zapewniony znacznie krótszy czas uruchomienia w przypadku, gdy potrzebują wymiany modułów, ponieważ w większości przypadków moduł zostanie zaktualizowany zanim program się do niego odwoła.



mgr inż. Marek PASSOWICZ  
RZECZNIK PATENTOWY



### **Zastrzeżenia patentowe**

1. Sposób obsługi odbioru oprogramowania urządzenia, według którego oprogramowanie to nadaje się w strumieniu danych, w postaci karuzeli obiektów, zaś po odbiorze i dekompresji, oprogramowanie zapisuje się w pamięci urządzenia, **znamienny tym, że** od momentu podłączenia odbiornika do strumienia danych pobiera się należące do określonej karuzeli dane, w kolejności nadawania tych danych w strumieniu, a po odebraniu wszystkich danych tworzy się z nich moduły, które się dekompresuje, a następnie dołącza ich zawartość do systemu plików urządzenia.
2. Sposób według zastrz.1, **znamienny tym, że** z pakietów strumienia danych pobiera się sekcję pakietu zawierającą moduły karuzeli, po czym sprawdza się czy sekcja zawiera potrzebne dane, które zapisuje się w pamięci urządzenia, sprawdzając przy tym kompletność pobranej karuzeli obiektów.
3. Sposób według zastrz. 1, **znamienny tym, że** pobierane ze strumienia danych sekcje danych, związane z określoną karuzelą obiektów, nadaje się w pakietach o określonym numerze PID.
4. Sposób według zastrz. 3, **znamienny tym, że** odbiornik podłącza się do strumienia danych w odpowiedzi na żądanie działającej w odbiorniku aplikacji, określające numer PID pakietów, w których jest nadawana żądana przez aplikację karuzela obiektów.

5. Sposób według zastrz.4, **znamienny tym, że** odbiornik podłącza się do strumienia danych w przypadku, gdy menedżer aplikacji chce uruchomić aplikację, która jest sygnalizowana w danym serwisie wraz z numerem PID pakietów, w których nadawana jest karuzela obiektów sygnalizowanej aplikacji.



mgr inż. Marek PASSOWICZ  
RZECZNIK PATENTOWY

1 / 5

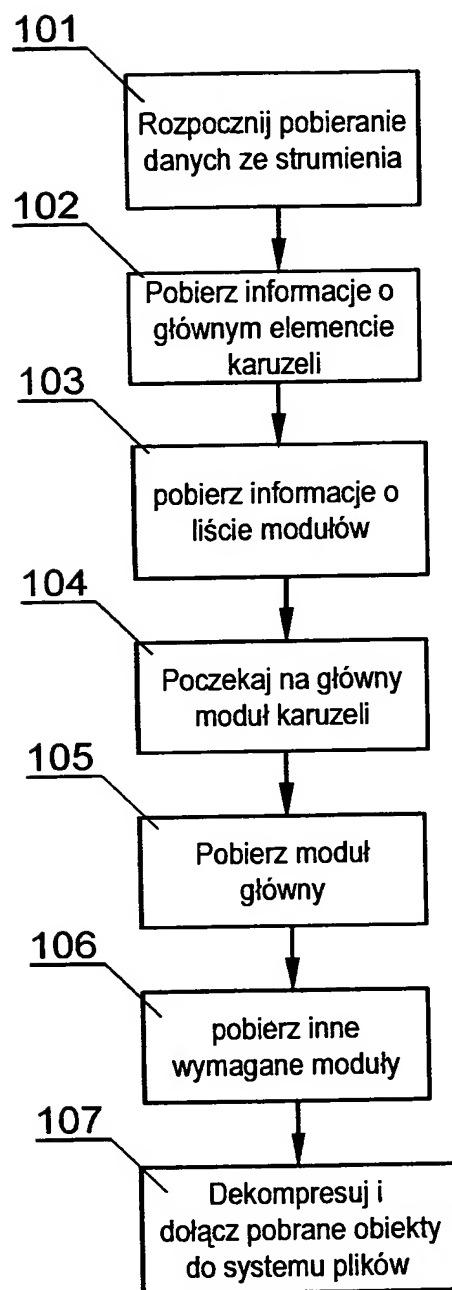


Fig. 1

2 / 5

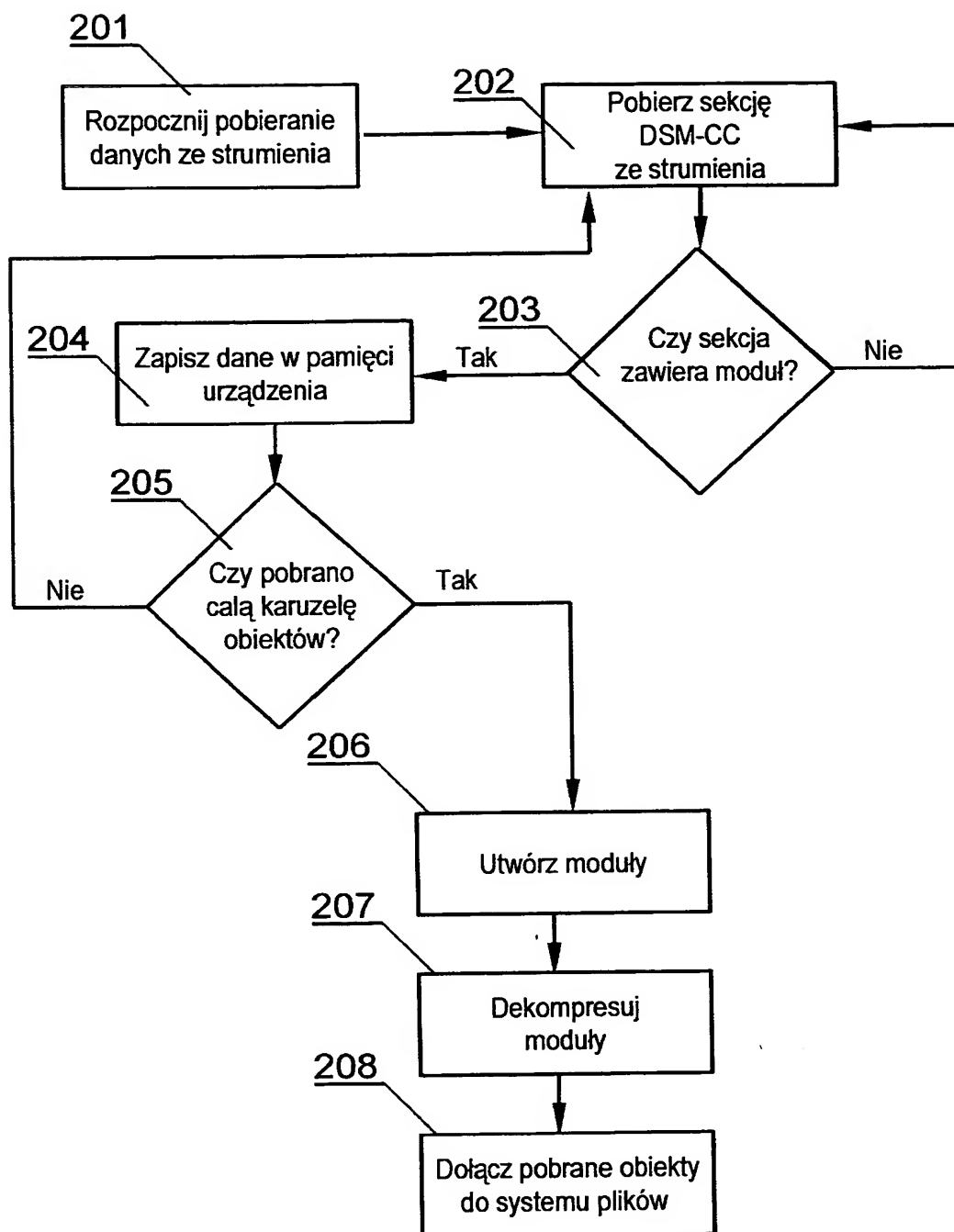


Fig. 2

3 / 5

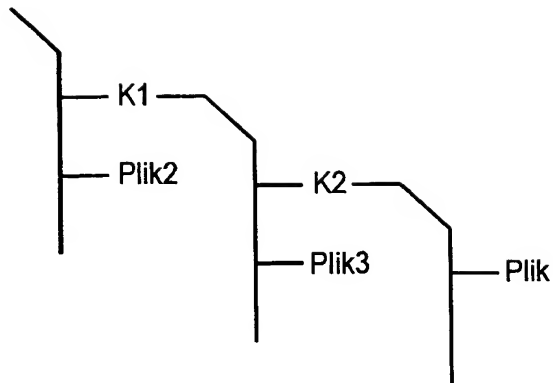


Fig. 3A

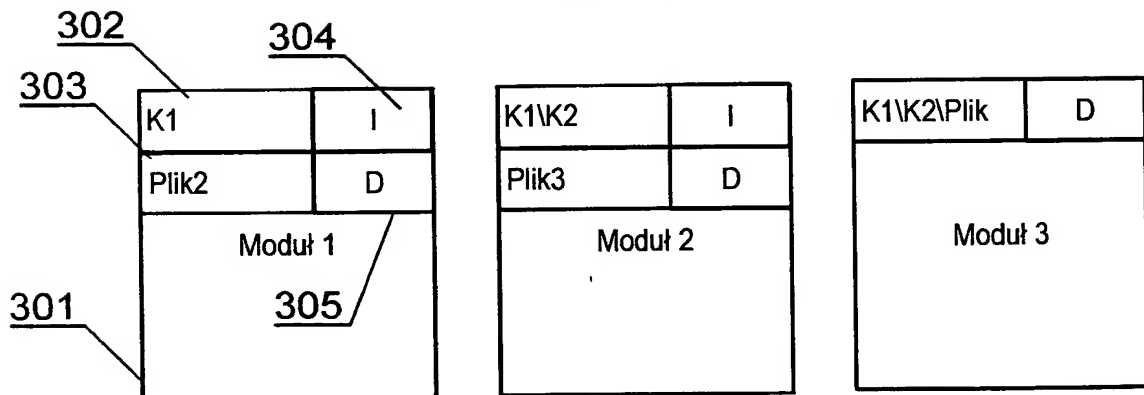


Fig. 3B

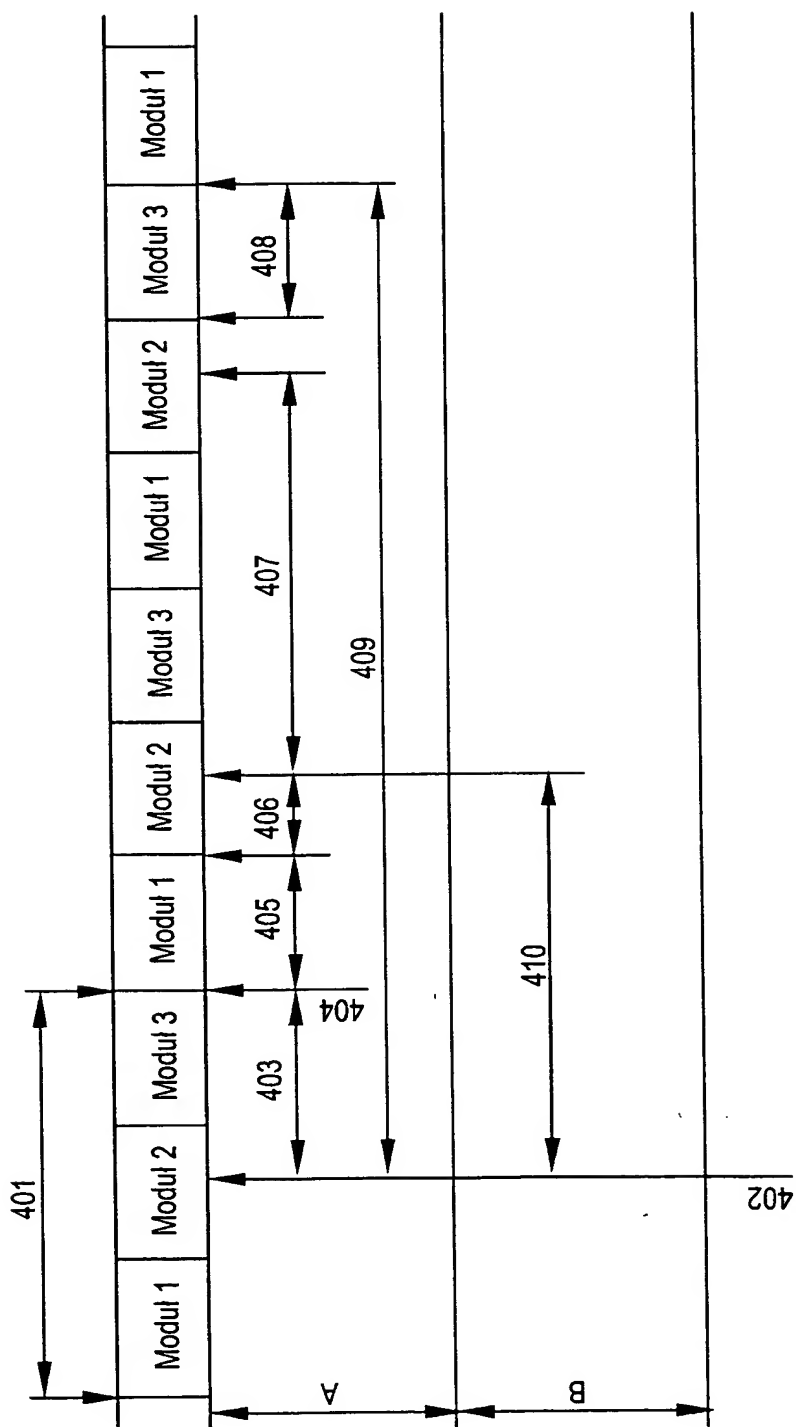


Fig. 4

5 / 5

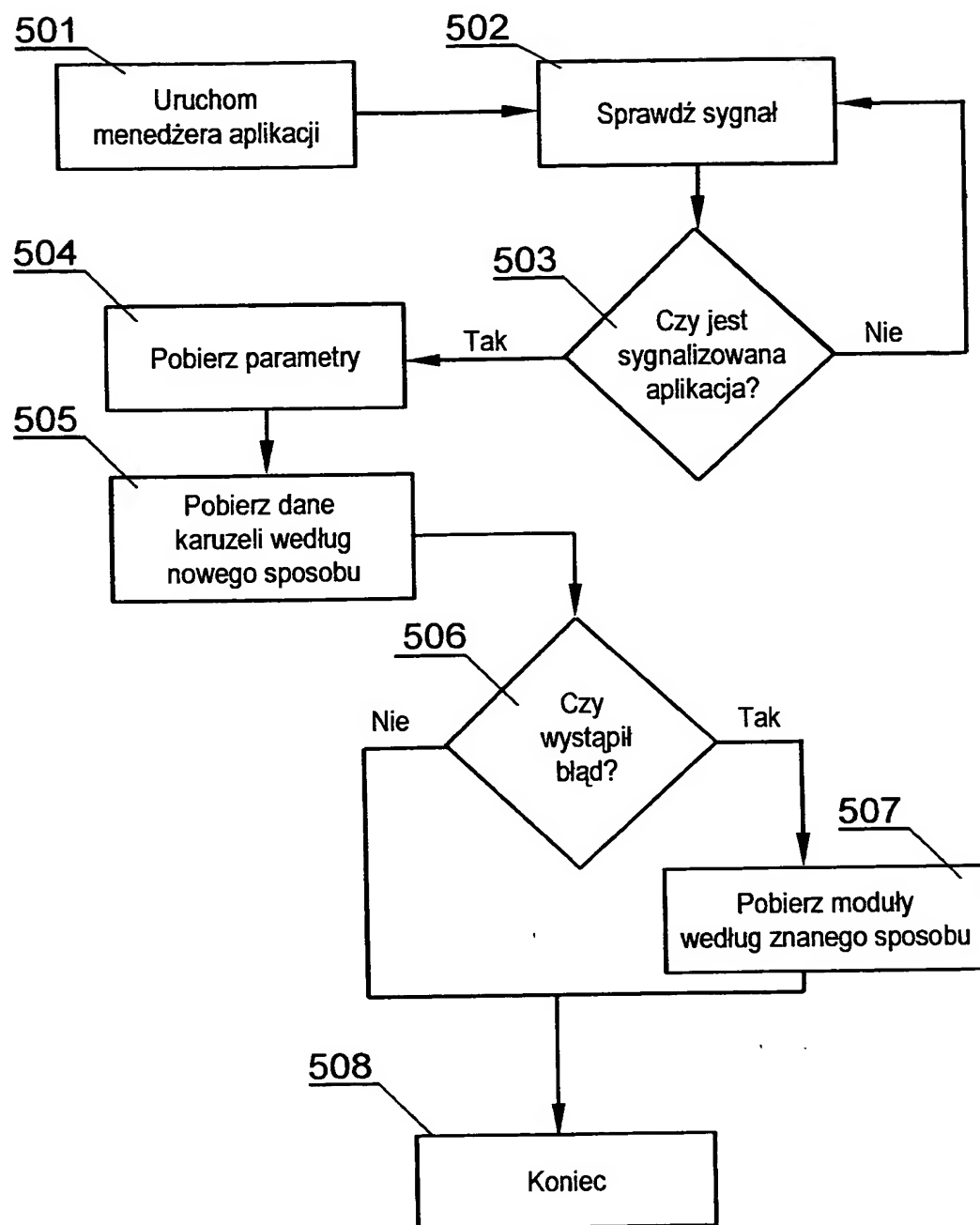


Fig. 5

**Translation from the Polish language**

**THE PATENT OFFICE OF THE REPUBLIC OF POLAND**

*/in the middle of the page the national emblem of the Republic of Poland/*

**A CERTIFICATE**

Advanced Digital Broadcast Ltd.,  
Taipei, Taiwan

Advanced Digital Broadcast Polska Sp.z o.o.,  
Zielona Góra, Poland

on February 10<sup>th</sup> 2003 submitted to the Patent Office of the Republic of Poland an application for granting a patent for an invention called „**A method of servicing reception of software broadcasted in a carousel of objects**”.

The description of the invention, the patent claims and the drawings, which were attached to this certificate, are true copies of the documents, which were submitted together with the application on February 10<sup>th</sup> 2003.

The application was submitted under the following number: P-358659.





Warsaw, February 11, 2003

on behalf of the President  
Jowita Mazur, MA  
Specialist

*/-/ illegible signature*

*/in the left hand lower corner of the page the round impressed golden sticker with the national emblem of the Republic of Poland in the middle and the following inscription in the rim: /*

PATENT OFFICE

\* 1 \*

*/the content of the sticker impressed at the bottom of each of the pages of the document /*

*/in the right hand upper corner of the page the following number: /*

358659

*/the number in handwriting: /*

4

## **A method of servicing reception of software broadcasted in a carousel of objects**

The object of the invention is a method of receiving software broadcasted in a carousel of objects, especially to manage reception of software in digital TV decoders, operating under control of MHP (Multimedia Home Platform) system.

The known method of updating application in MHP system, assumes exchange of data only at a request of the program, i.e. by referring the application to the missing module. After such request the appropriate module is being searched for in the carousel of objects broadcasted in the stream. It should be emphasized that this is greatly time-consuming process.



The method of managing data of a carousel of objects, which is known from American patent application no. US 2002/0059645 assumes introduction of modification at the side of the transmitter. The writing of objects broadcasted in the carousel is conducted at the level of application modules, while only the version of the module and time, during which the given module is valid are stored in the device. Additionally if versions of modules change, all versions and time of their validity are stored in the module. The requirement of modifying the data stream at the side of the transmitter and accumulating excessive information in a receiver makes this way a very complicated one, and thus time consuming and inconvenient in use.

The essence of the method, according to the invention consists in that from the moment of connecting a receiver to the data stream, broadcasting its software in the form of a carousel of objects; data belonging to this carousel are retrieved in the sequence of broadcasting these data in the stream. After all data are received, modules are created out of them. These modules are decompressed, and next their content is appended to the file system of the device.

It is favorable when a section of the packet including carousel modules is retrieved from the data stream packets. Next it is checked if the section includes required data, which are stored in the storage of the device while completeness of the retrieved carousel of objects is checked.

More over it is favorable when the data sections, retrieved from the data stream, which are associated with a specific carousel of objects, are broadcasted in packets with a specific PID number.

It is also favorable when a receiver is connected to the data stream in an answer to a request of the application active in a receiver. The request defines the number of PID packets, in which the carousel of objects requested by the application is broadcasted.

In the same time it is favorable when a receiver is connected to the data stream in case, the applications manager wants to start the application, which is signaled in a given service together with a number of PID packets, in which the carousel of objects of the signaled application is broadcasted.

The presented method allows capturing information broadcasted cyclically in the carousel of objects and creating a file system, which is later used by applications, operating for example



under control of MHP system. However the data modules are not retrieved only as a result of reaction to referring the application to the missing module, but also immediately after finding a change of data broadcasted in the carousel of objects. The solution does not require any modifications of the data stream at the side of the transmitter and acts without additional information related to validity of the broadcasted modules. Moreover only the most recent version of the module i.e. its identifier and the module itself are stored in a receiver.

The object of the invention was described in detail in the example, while for better understanding of it a drawing was used, and its separate figures represent the following:

- Fig. 1 – the method of receiving the program known from the prior art,
- Fig. 2 – the method of receiving the program according to the invention,
- Fig. 3 – the method of organizing file system by means of modules,
- Fig. 4 – a comparison of the time of data retrieving performed according to the known method and the method of the invention,
- Fig. 5 – the use of the method in case certain modules were not retrieved earlier.

**Example.** The broadcaster sends software together with the audio/video signal in one data stream. A receiver, which can be a digital television decoder IRD (Integrated Receiver Decoder), receives the software, which is next stored in the storage of a receiver, and subsequently manually started in the device. Such exemplary software is EPG - Electronic Program Guide, games or news services. According to the prior art, applications are broadcasted in form of modules in the carousel of objects, where data and the software code are broadcasted cyclically. Such solution resembles the teletext system, in which pages are broadcasted in cycles.

The exemplary system, in which transmission in the television data stream takes place, is known from international standard ISO/IEC 13818-6 – MPEG-2 Digital Storage Media Command and Control – DSM-C. The MPEG stream includes multiplexed television channels. Any such television channel can include interactive applications designated to a receiver.



Objects, which are sent by means of DSM-CC protocol, are a part of the carousel of objects. They are broadcasted in the form of modules, which create a system of objects, files. The organization of the system of files stored at the side of a receiver is similar to the system, known from standard PC computers. When a receiver, for example, a digital television decoder wants to retrieve an object from a carousel of objects, it must know, in which module the given object is located. After appropriate module is retrieved by a receiver, the module is analyzed in order to find the right object. Organization of modules was presented in **Fig. 3B**, where three modules include a system of files described in **Fig. 3A**. Each module (301) can include files (303) and directories (302). If the module includes a file, it must also include its content (305), while in case of a directory the module includes information (304) about files/directories, which are located in it. This information includes the name of the file/directory and, among others, information about the module, in which this file/directory is broadcasted.

Due to the hierarchy of objects broadcasted in the carousel, the given object can be located in one of sub-directories. In such case, in order to find the object, a receiver must also collect all superior modules in relation to the module storing the object, which must be analyzed before a receiver reads the appropriate object. For example, if a receiver needs to retrieve the object "File" located in path "K1/K2/File", it must check first in which module object "K1" is broadcasted, next retrieve object "K1", read in which module object "K2" is broadcasted, possibly retrieve this module. And subsequently read from object "K2", in which module object "File" is broadcasted. In the simplest case, all three objects can be in one module and then it is enough to retrieve only this single module. In the most complicated case, which is illustrated in **Fig. 3b**, each of these objects is in a different module. In order to check if the path is correct and the object "File" exists, all three modules must be retrieved from the data stream.

Most frequently the carousel of objects is sent in a compressed form. Compression is added at the level of modules. As a result a receiver has additional difficulty, which is the requirement of decompressing all the modules required to find the objects, which at a given time are required by a receiver. A single module includes usually many objects, which are compressed as a whole. Such system, considering a possibility of data processing in receivers, like for example digital television receivers, is time consuming and demands high system resources.



Additional hindrance is the fact that files and directories of the system of the carousel of objects can change in time. In the DSM-CC data transmission system the version is defined only at the level of modules, and not at the level of the hierarchy of directories or the objects themselves. In case only one object from the module changes, the whole module will receive a new version and must be once again retrieved as a whole from the carousel of objects.

Another impediment is the fact that objects can be shifted among separate modules. They can also be added or removed. Such situation leads to quite frequent exchanges of objects, which constitute a part of the software of a receiver, such as for example digital television decoder. In the presented method the object with changed location will be retrieved faster.

According to the typical method of data reception, in order to append the carousel of objects to the file system of a receiver, typically digital television receiver, first the module with the main element of the carousel should be retrieved. Next the consecutive modules are retrieved, depending on what objects are needed by the application. The typical algorithm of data retrieving looks like it was illustrated in Fig. 1. After connecting to the data stream (101), information about the main module (root) is retrieved in point (102) of the procedure. Next information about the list of modules, broadcasted in the carousel (103), is retrieved. After analyzing this information one should wait by the time when the main module (root) (104) is broadcasted. The next point is to retrieve the main module (root) of the carousel of objects (105). The next step of the procedure is to read other modules (106) needed by the application. The last task of the procedure is to append the retrieved objects to the file system (107), which is written in the storage of the receiver (usually it is a FLASH type memory), for example a decoder of digital television.

In order to define the time needed for correct retrieving of the update for the application, which includes five modules, the following calculations should be made:

- It is assumed that retrieving information about the main module (root) of the carousel lasts for time  $t_{\text{root}}$ ,
- It is assumed that the average time of waiting for information about the module (root) of carousel lasts for time  $t_{\text{root\_wait}}$ ,
- It is assumed that retrieving information about the lists of modules lasts for time  $t_{\text{list}}$ ,



- It is assumed that the average waiting time for information about the list of modules lasts  $t_{list\_wait}$ ,
- It is assumed that retrieving one module lasts time  $t_{module}$ ,
- It is assumed that the average waiting time before the module starts to be broadcasted is  $t_{module\_wait}$ .

In case the application needs  $N$  modules for operation, the total time of retrieving the modules for this application is:

$$t_{root} + t_{root\_wait} + t_{list} + t_{list\_wait} + N * (t_{module\_wait} + t_{module})$$

It is assumed in the proposed method that in most cases the application needs all the modules, which are broadcasted in the carousel of objects. Moreover the carousel is broadcasted cyclically, without longer breaks between its end and beginning.

Since the application usually needs all the modules from the carousel of objects, it is not favorable to wait until the moment when the program refers to the nonexistent module in the system, but all modules should be retrieved at once. In such case one eliminates the waiting time for separate information, and all the modules start to be retrieved immediately from the moment of connecting the receiver to the data stream. Instead of retrieving the broadcasted modules in appropriate sequence, all modules are retrieved in the sequence, in which they are broadcasted in the carousel of objects. It is expected that these are correct data and information about them will be broadcasted very soon. Therefore, it may happen that first the modules and next the information about them will be retrieved. The time of retrieving the same application, assuming that there are  $N$  modules in the carousel, is:  $N * t_{module}$ .

Therefore the average time gained is:

$$t_{root} + t_{root\_wait} + t_{list} + t_{list\_wait} + N * t_{module\_wait}$$

Information about the list of modules and the main module (root) is broadcasted between modules with a much higher frequency, thus when calculating the time of retrieving application, using the method according to the invention, one does not have to add times  $t_{root\_wait} + t_{list\_wait} + t_{list}$ . However, in certain cases the method, according to the invention, might have worse results regarding the time of retrieving the modules. For



example, in case when there are many modules in the carousel and the application needs only one.


There is also risk that a few of the modules will not be retrieved earlier and in such case the other required modules are retrieved according to the typical method. Such situation was presented in **Fig. 5**. In point (501) applications manager, managing the programs of a given device is started. After the start, the data sent in the stream, for example MPEG2 format data, are monitored (502). It is checked sequentially if the signaled application (503) is in the stream. If there is not such signal, the data sent in the stream are monitored further. In opposite case, information defining the parameters of the application, broadcasted in the application (504) stream, is retrieved. The next step of the procedure (505) is to retrieve the carousel of objects, according to the method shown in **Fig. 2**. If during retrieving the data an error (506) occurs, the modules are retrieved in the step (507) from the carousel according to the known methods – **Fig. 1**. Errors include situations when the receiver has too low memory to write the whole carousel of objects. Other problems are, for example: a decrease of the signal quality or a situation when during retrieving a given module its version changes.

Data retrieving is performed continually by the time when the receiver finds that all data from the carousel broadcasted in the stream packets with a specific PID number were already retrieved. PID number is defined by application, which requests data reading from the carousel, broadcasted in the elementary stream packets with a specific PID number. Continual retrieving is understood in such a way that a filter of section DSM-CC of MPEG2 packets is activated thus it passes all possible data connected with the carousel broadcasted in the packets with a specific PID number. Only later, modules and other information are compiled from the retrieved data. If data were to be retrieved module by module, the filter would have to be set to consecutive modules, and next it would have to be switched off and switched on again, in order to retrieve the next module. During the time when the filter is turned off, the module, which was to be retrieved, can already be broadcasted, which would be result in waiting for the next broadcast of the carousel of objects. Additionally, because the sequence of broadcasting modules is not known there is never a certainty that they would be retrieved in appropriate, optimum sequence. The method, according to the invention, eliminates these problems and the losses of time, thanks to retrieving all data sections, as they are broadcasted in the carousel. Next their real content is checked. The method of retrieving data from the stream, for example in MPEG-2 format was illustrated in **Fig. 2**. The procedure starts with the



time of connecting to the data stream (201), which may be a result of a request from the side of the application. Next, sections DSM-CC are retrieved from packets with a specific PID number (202). The next stage (203) is to check if the retrieved data section includes the module. If this is a fact, a shift to point (204) of the procedure is made. Otherwise a return to point (202) is made, where the next section from the stream is received. In the place marked with number (204) data are stored in the memory of the device. Next it is checked if all data sections (205), broadcasted in the carousel of objects with the use of protocol DSM-CC, were already retrieved. If not, the procedure returns to point (202). In opposite case, modules will be created from the retrieved data (206). The next step is decompression of modules (207), so that appropriate objects broadcasted in the carousel are received in effect. The last task of the procedure is to place the retrieved objects in the file system (208), which is written in the memory of the receiver – usually this is a FLASH type memory, for example a digital television decoder.

As practical tests show, the method according to the invention guarantees on average a twice-faster delivery of the new module to the application operating in the receiver, for example a digital television decoder. An exemplary comparison of the time needed to retrieve the carousel of objects consisting of three modules was illustrated in Fig. 4. Part A shows the method known from the state of technology, while part B – the method according to the invention. One cycle of the carousel of objects was marked with a number (402). Proceeding in line with the method A, one should wait until the main module (root) appears in the stream, marked with symbol “Module 1”. This time is marked with a number (403). In point (404) retrieving of the main module starts, which lasts for the time, defined with the number (405). The time (406) means a delay resulting from analysis of the main module by the application. Due to this delay, retrieving of the second module starts already when a part of “Module 2” is broadcasted. That is why in order to retrieve the rest of “Module 2” one should wait until it is broadcasted once again. The time of retrieving “Module 2” is defined by the range (407). “Module 3” is retrieved as the last one, in the time indicated with a number (408). The whole operation of retrieving the three modules, according to the known method, lasts for the time marked with the number (409). According to the invention, the modules are retrieved from the stream one by one, as they are broadcasted. Immediately after connecting to the data stream, modules are retrieved by the time when all the modules are retrieved. The whole operation occurs in the time marked as (410). Only later they are analyzed. Thanks to that, the time needed to update the data is reduced to the indispensable minimum and a much shorter start-up time is ensured for the applications operating in the receiver, in case they need an





exchange of modules. Since, in majority of cases the module will be updated before the program refers to it.

*/oblong stamp with the following contents:/*

Eng. Marek PASSOWICZ, MA

Patent Attorney

*/-/ illegible signature*

*/in the right hand upper corner of the page the following number:/*

358659

*/the number in handwriting:/*

5

### **Patent claims**

1. The method of handling reception of device software, according to which the software is broadcasted in the data stream, in the form of the carousel of objects, while after reception and decompression, the software is written in the device storage, the method **is characterized in that** from the moment of connecting the receiver to the data stream, the data belonging to the defined carousel are retrieved, in the sequence of broadcast of these data in the stream. After receiving all data, modules are created out of them and then these modules are decompressed, and next their content is appended to the file system of the device.
2. The method, according to claim 1, **characterized in that**, from the packet of the data stream a section of the packet is retrieved, which includes the module of the carousel, next it is checked if the section includes the needed data, which are written in the storage of the device, at the same time the completeness of the retrieved carousel of objects is checked.
3. The method, according to claim 1, **characterized in that** the data sections, retrieved from the data stream, related to the defined carousel of objects, are broadcasted in packets with a specific PID number.



4. The method, according to claim 3, **characterized in that** the receiver is being connected to the data stream in an answer to the request of the application operating in the receiver, the request defines the number of PID packets, in which the carousel of objects requested by the application is broadcasted.
5. The method, according to claim 4, **characterized in that** the receiver is being connected to the data stream in case when the applications manager wants to start the application, which is signaled in a given service together with the PID number of packets, in which the carousel of objects of the signaled application is broadcasted.

*/oblong stamp with the following contents:/*

Eng. Marek PASSOWICZ, MA

PATENT ATTORNEY

*/-/ illegible signature*

*/on the subsequent pages in the right hand corner the following number is repeated:/*

358659

*/the subsequent pages are numbered in handwriting from 6 to 10/*

*/the subsequent pages are stamped at the bottom of the page with oblong stamps with the following contents:/*

Eng. Marek PASSOWICZ, MA

PATENT ATTORNEY

*/-/ illegible signature*

A handwritten signature in black ink, located in the bottom right corner of the page. The signature is stylized and appears to be the name of the patent attorney, Marek Passowicz.

1 / 5

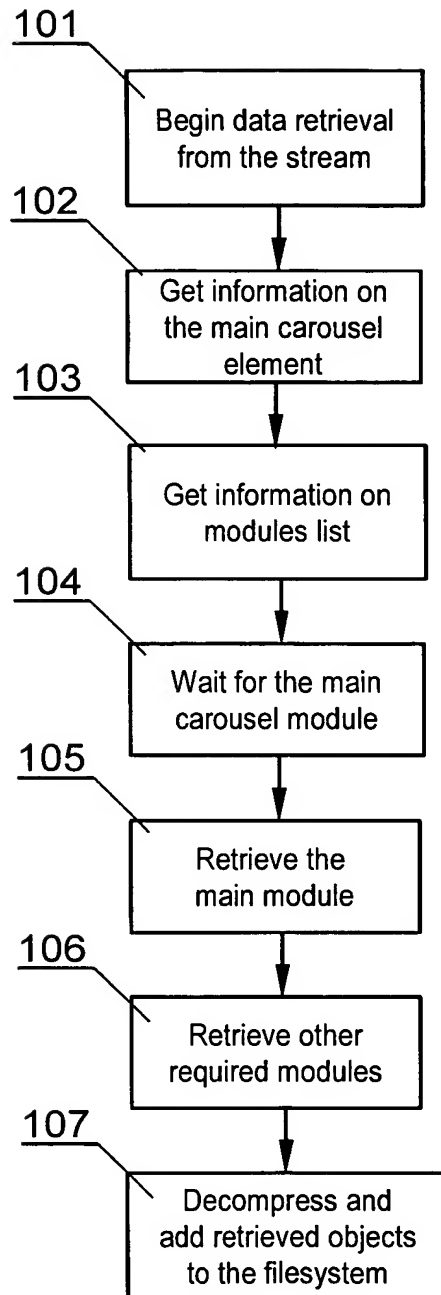


Fig. 1

*Handwritten signature*

2 / 5

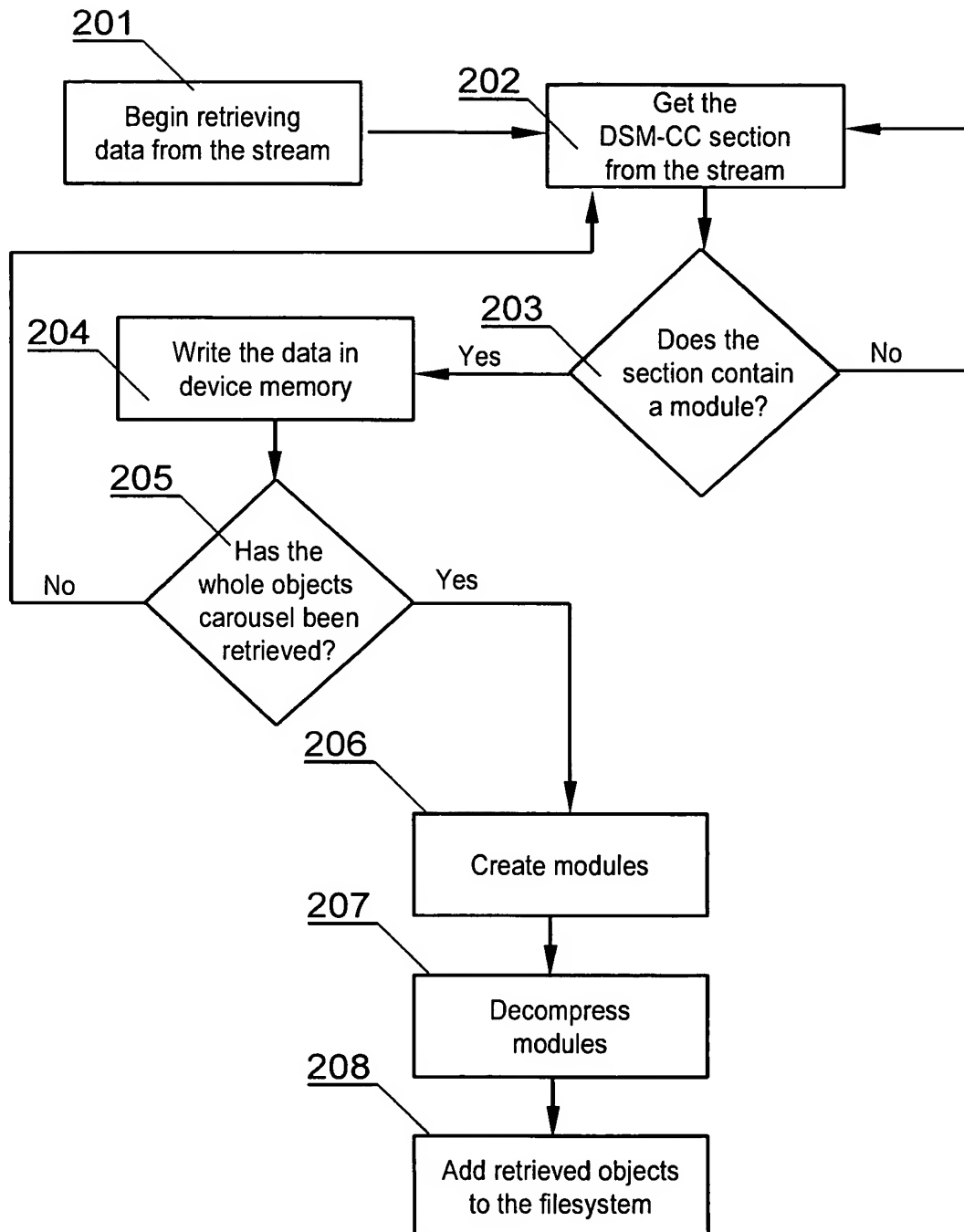


Fig. 2

*Blue*

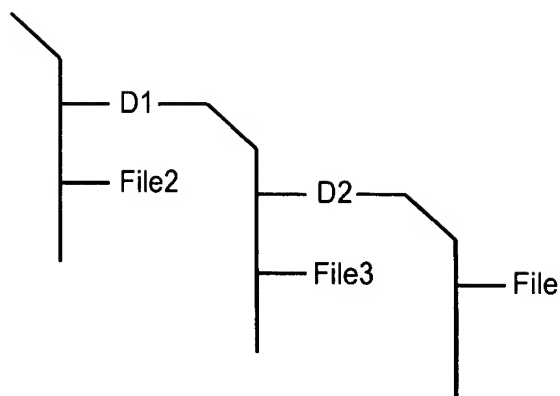


Fig. 3A

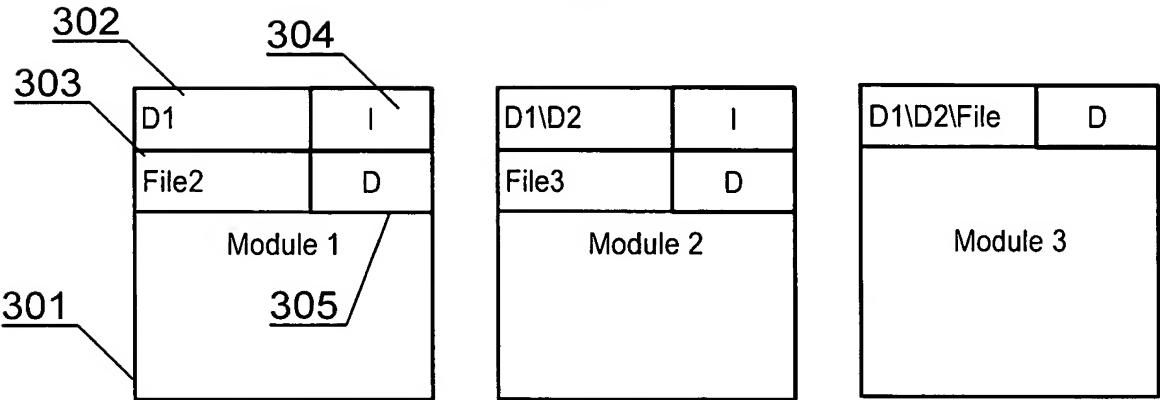


Fig. 3B

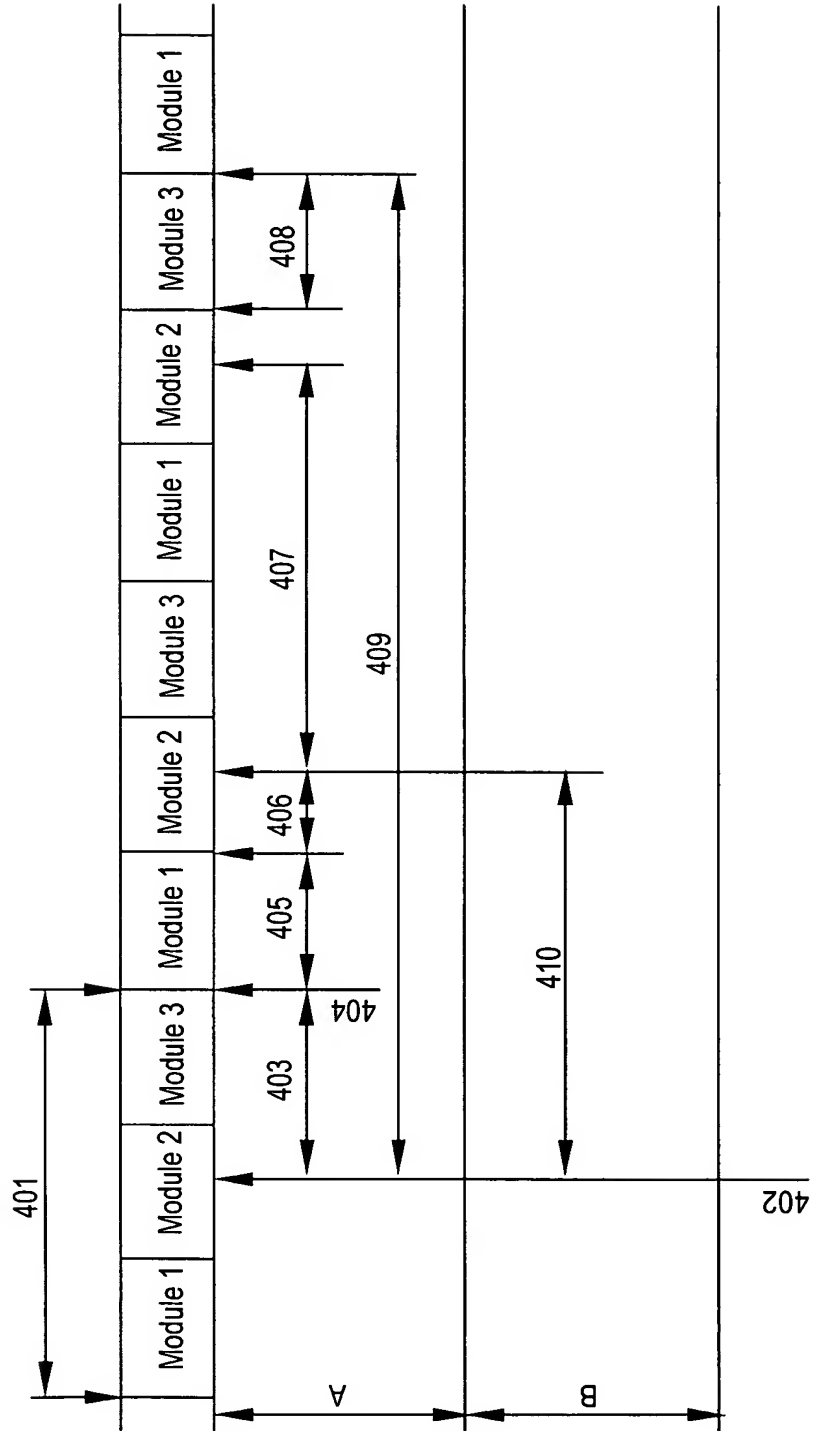


Fig. 4

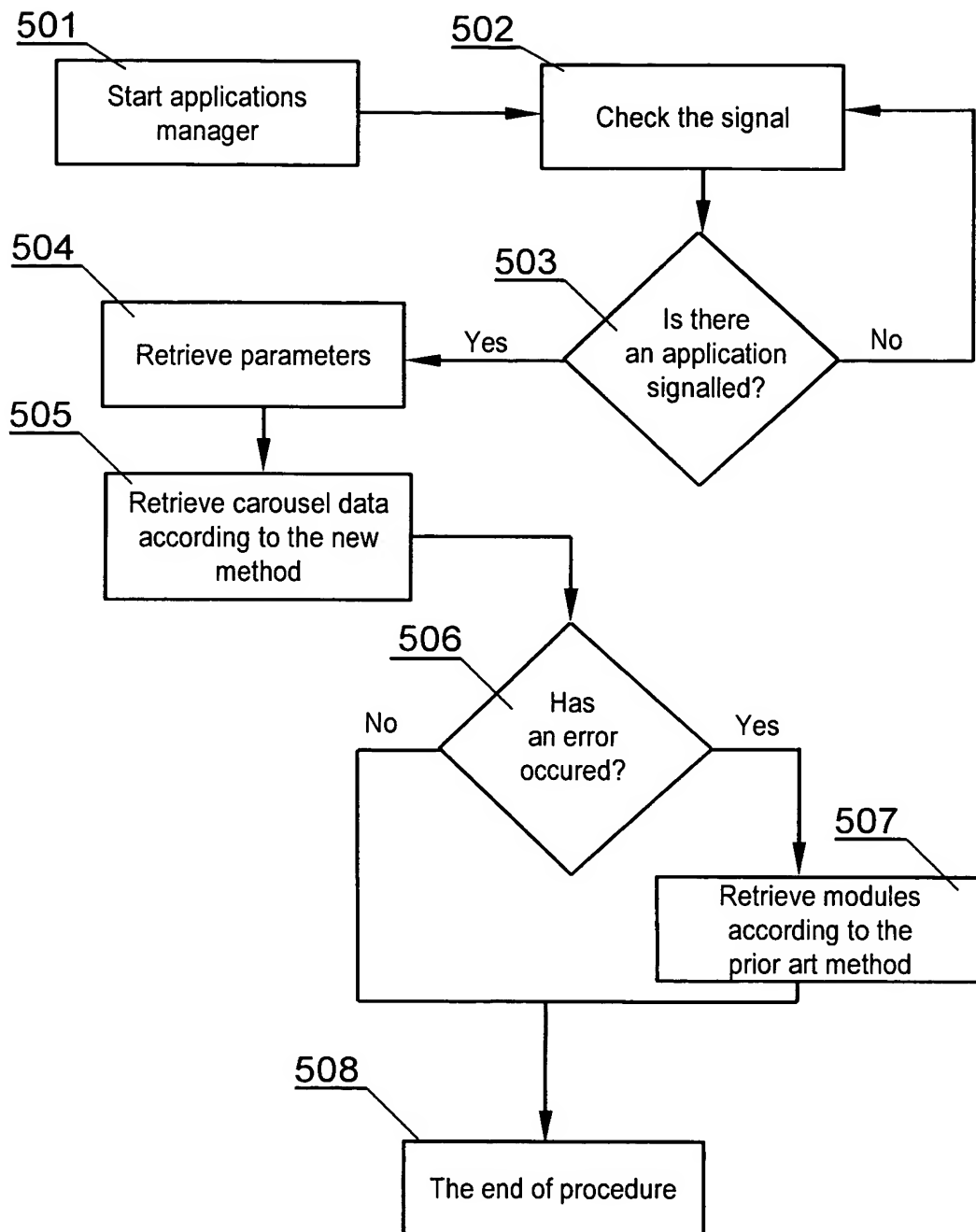


Fig. 5

---

**Repertory No.: 140/3/2004**

*I, the undersigned, Iwona Duma, sworn translator of the English language for the District Court of the City of Warsaw, hereby certify that the above text is a true and complete translation of the Polish original document presented to me.*

*Warsaw, March 3, 2004.*

A handwritten signature in black ink, appearing to read 'Iwona Duma', with a large, stylized initial 'I'.A second handwritten signature in black ink, identical to the one above, located in the bottom right corner of the page.